**BERT-ATTACK: Adversarial Attack Against BERT Using BERT**

# Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, Xipeng Qiu*\**

Shanghai Key Laboratory of Intelligent Information Processing, Fudan University School of Computer Science, Fudan University

825 Zhangheng Road, Shanghai, China

*(*linyangli19,rtma19,qpguo16,xyxue,xpqiu*}*@fudan.edu.cn

# Abstract

Adversarial attacks for discrete data (such as texts) have been proved signiﬁcantly more challenging than continuous data (such as im- ages) since it is difﬁcult to generate adversar- ial samples with gradient-based methods. Cur- rent successful attack methods for texts usually adopt heuristic replacement strategies on the character or word level, which remains chal- lenging to ﬁnd the optimal solution in the mas- sive space of possible combinations of replace- ments while preserving semantic consistency and language ﬂuency. In this paper, we pro- pose **BERT-Attack**, a high-quality and effec- tive method to generate adversarial samples using pre-trained masked language models ex- empliﬁed by BERT. We turn BERT against its ﬁne-tuned models and other deep neural mod- els in downstream tasks so that we can success- fully mislead the target models to predict incor- rectly. Our method outperforms state-of-the- art attack strategies in both success rate and perturb percentage, while the generated adver- sarial samples are ﬂuent and semantically pre- served. Also, the cost of calculation is low, thus possible for large-scale generations. The code is available at [https://github.com/](https://github.com/LinyangLee/BERT-Attack) [LinyangLee/BERT-Attack](https://github.com/LinyangLee/BERT-Attack).

# Introduction

Despite the success of deep learning, recent works have found that these neural networks are vulnera- ble to adversarial samples, which are crafted with small perturbations to the original inputs ([Goodfel-](#_bookmark25) [low et al.](#_bookmark25), [2014](#_bookmark25); [Kurakin et al.](#_bookmark28), [2016](#_bookmark28); [Chakraborty](#_bookmark21) [et al.](#_bookmark21), [2018](#_bookmark21)). That is, these adversarial samples are imperceptible to human judges while they can mis- lead the neural networks to incorrect predictions. Therefore, it is essential to explore these adver- sarial attack methods since the ultimate goal is to make sure the neural networks are highly reliable

*\**Corresponding author.

and robust. While in computer vision ﬁelds, both attack strategies and their defense countermeasures are well-explored ([Chakraborty et al.](#_bookmark21), [2018](#_bookmark21)), the adversarial attack for text is still challenging due to the discrete nature of languages. Generating of adversarial samples for texts needs to possess such qualities: (1) imperceptible to human judges yet misleading to neural models; (2) ﬂuent in grammar and semantically consistent with original inputs.

Previous methods craft adversarial samples mainly based on speciﬁc rules ([Li et al.](#_bookmark30), [2018](#_bookmark30); [Gao](#_bookmark23) [et al.](#_bookmark23), [2018](#_bookmark23); [Yang et al.](#_bookmark40), [2018](#_bookmark40); [Alzantot et al.](#_bookmark18), [2018](#_bookmark18); [Ren et al.](#_bookmark37), [2019](#_bookmark37); [Jin et al.](#_bookmark27), [2019](#_bookmark27); [Zang et al.](#_bookmark41), [2020](#_bookmark41)). Therefore, these methods are difﬁcult to guaran- tee the ﬂuency and semantically preservation in the generated adversarial samples at the same time. Plus, these manual craft methods are rather com- plicated. They use multiple linguistic constraints like NER tagging or POS tagging. Introducing contextualized language models to serve as an au- tomatic perturbation generator could make these rules designing much easier.

The recent rise of pre-trained language models, such as BERT ([Devlin et al.](#_bookmark22), [2018](#_bookmark22)), push the per- formances of NLP tasks to a new level. On the one hand, the powerful ability of a ﬁne-tuned BERT on downstream tasks makes it more challenging to be adversarial attacked ([Jin et al.](#_bookmark27), [2019](#_bookmark27)). On the other hand, BERT is a pre-trained masked language model on extremely large-scale unsupervised data and has learned general-purpose language knowl- edge. Therefore, BERT has the potential to gener- ate more ﬂuent and semantic-consistent substitu- tions for an input text. Naturally, both the proper- ties of BERT motivate us to explore the possibility of attacking a ﬁne-tuned BERT with another BERT as the attacker.

In this paper, we propose an effective and high-quality adversarial sample generation method: **BERT-Attack**, using BERT as a language model

6193

*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6193–6202, November 16–20, 2020. *O*c 2020 Association for Computational Linguistics

to generate adversarial samples. The core algo- rithm of BERT-Attack is straightforward and con- sists of two stages: ﬁnding the vulnerable words in one given input sequence for the target model; then applying BERT in a semantic-preserving way to generate substitutes for the vulnerable words. With the ability of BERT, the perturbations are generated considering the context around. There- fore, the perturbations are ﬂuent and reasonable. We use the masked language model as a perturba- tion generator and ﬁnd perturbations that maximize the risk of making wrong predictions ([Goodfellow](#_bookmark25) [et al.](#_bookmark25), [2014](#_bookmark25)). Differently from previous attacking strategies that require traditional single-direction language models as a constraint, we only need to in- ference the language model once as a perturbation generator rather than repeatedly using language models to score the generated adversarial samples in a trial and error process.

Experimental results show that the proposed BERT-Attack method successfully fooled its ﬁne- tuned downstream model with the highest attack success rate compared with previous methods. Meanwhile, the perturb percentage and the query number are considerably lower, while the semantic preservation is high.

To summarize our main contributions:

We propose a simple and effective method, named **BERT-Attack**, to effectively generate ﬂuent and semantically-preserved adversarial samples that can successfully mislead state- of-the-art models in NLP, such as ﬁne-tuned BERT for various downstream tasks.

*●*

BERT-Attack has a higher attacking success rate and a lower perturb percentage with fewer access numbers to the target model compared with previous attacking algorithms, while does not require extra scoring models there- fore extremely effective.

*●*

# Related Work

To explore the robustness of neural networks, adver- sarial attacks have been extensively studied for con- tinuous data (such as images) ([Goodfellow et al.](#_bookmark25), [2014](#_bookmark25); [Nguyen et al.](#_bookmark34), [2015](#_bookmark34); [Chakraborty et al.](#_bookmark21), [2018](#_bookmark21)). The key idea is to ﬁnd a minimal pertur- bation that maximizes the risk of making wrong predictions. This minimax problem can be eas- ily achieved by applying gradient descent over the continuous space of images ([Miyato et al.](#_bookmark32), [2017](#_bookmark32)).

However, adversarial attack for discrete data such as text remains challenging.

## Adversarial Attack for Text

Current successful attacks for text usually adopt heuristic rules to modify the characters of a word ([Jin et al.](#_bookmark27), [2019](#_bookmark27)), and substituting words with syn- onyms ([Ren et al.](#_bookmark37), [2019](#_bookmark37)). [Li et al.](#_bookmark30) ([2018](#_bookmark30)); [Gao](#_bookmark23) [et al.](#_bookmark23) ([2018](#_bookmark23)) apply perturbations based on word em- beddings such as Glove ([Pennington et al.](#_bookmark35), [2014](#_bookmark35)), which is not strictly semantically and grammati- cally coordinated. [Alzantot et al.](#_bookmark18) ([2018](#_bookmark18)) adopts lan- guage models to score the perturbations generated by searching for close meaning words in the word embedding space ([Mrksˇic´ et al.](#_bookmark33), [2016](#_bookmark33)), using a trial and error process to ﬁnd possible perturbations, yet the perturbations generated are still not context- aware and heavily rely on cosine similarity mea- surement of word embeddings. Glove embeddings do not guarantee similar vector space with cosine similarity distance, therefore the perturbations are less semantically consistent. [Jin et al.](#_bookmark27) ([2019](#_bookmark27)) apply a semantically enhanced embedding ([Mrksˇic´ et al.](#_bookmark33), [2016](#_bookmark33)), which is context unaware, thus less consis- tent with the unperturbed inputs. [Liang et al.](#_bookmark31) ([2017](#_bookmark31)) use phrase-level insertion and deletion, which pro- duces unnatural sentences inconsistent with the original inputs, lacking ﬂuency control. To pre- serve semantic information, [Glockner et al.](#_bookmark24) ([2018](#_bookmark24)) replace words manually to break the language in- ference system ([Bowman et al.](#_bookmark19), [2015](#_bookmark19)). [Jia and](#_bookmark26) [Liang](#_bookmark26) ([2017](#_bookmark26)) propose manual craft methods to at- tack machine reading comprehension systems. [Lei](#_bookmark29) [et al.](#_bookmark29) ([2019](#_bookmark29)) introduce replacement strategies using embedding transition.

Although the above approaches have achieved good results, there is still much room for improve- ment regarding the perturbed percentage, attacking success rate, grammatical correctness and semantic consistency, etc. Moreover, the substitution strate- gies of these approaches are usually non-trivial, resulting in that they are limited to speciﬁc tasks.

## Adversarial Attack against BERT

Pre-trained language models have become main- stream for many NLP tasks. Works such as ([Wal-](#_bookmark38) [lace et al.](#_bookmark38), [2019](#_bookmark38); [Jin et al.](#_bookmark27), [2019](#_bookmark27); [Pruthi et al.](#_bookmark36), [2019](#_bookmark36)) have explored these pre-trained language models from many different angles. [Wallace et al.](#_bookmark38) ([2019](#_bookmark38)) explored the possible ethical problems of learned knowledge in pre-trained models.

# BERT-Attack

Motivated by the interesting idea of turning BERT against BERT, we propose **BERT-Attack**, using the original BERT model to craft adversarial sam- ples to fool the ﬁne-tuned BERT model.

Our method consists of two steps: (1) ﬁnding the vulnerable words for the target model and then

(2) replacing them with the semantically similar and grammatically correct words until a successful attack.

The most-vulnerable words are the keywords that help the target model make judgments. Pertur- bations over these words can be most beneﬁcial in crafting adversarial samples. After ﬁnding which words that we are aimed to replace, we use masked language models to generate perturbations based on the top-K predictions from the masked language model.

## Finding Vulnerable Words

Under the black-box scenario, the logit output by the target model (ﬁne-tuned BERT or other neural models) is the only supervision we can get. We ﬁrst select the words in the sequence which have a high signiﬁcance inﬂuence on the ﬁnal output logit. Let *S* = [*w*0*, , wi* ] denote the input sen- tence, and *oy*(*S*) denote the logit output by the target model for correct label *y*, the importance

*· · · · · ·*

score *Iwi* is deﬁned as

*Iwi* = *oy*(*S*) *. oy*(*S\wi* )*,* (1)

where *S\wi* = [*w*0*, , wi-*l*,* [MASK]*, wi*＋l*,* ]

*· · · · · ·*

is the sentence after replacing *wi* with [MASK].

Then we rank all the words according to the ranking score *Iwi* in descending order to create word list *L*. We only take *e* percent of the most im-

portant words since we tend to keep perturbations minimum.

This process maximizes the risk of making wrong predictions, which is previously done by cal- culating gradients in image domains. The problem is then formulated as replacing these most vulner- able words with semantically consistent perturba- tions.

## Word Replacement via BERT

After ﬁnding the vulnerable words, we iteratively replace the words in list *L* one by one to ﬁnd per- turbations that can mislead the target model. Previ- ous approaches usually use multiple human-crafted

**Generated Sample**

Figure 1: One step of our replacement strategy.

*w*

*w*

*w*



*pi*＋l

l

*pi*＋l

2

*pi*＋l

*k*

**Full-Permutation of top-K predictions**

*w w* **…** *c*l **…**

*p*l*i*

*p*2*i*

*pi*＋2

l

*pi*＋2

**Rank**

2

*k*

*k*

*c*l

*c*2

*ck*

**…**

*pi*

*pi*＋2

*wi c*2 *ck*

**BERT**

*w w* **… …** *wn wn*

subword of *wi*

**Input** *w w* **…** *wi* **…** *wn wn*

*c*l *c*2 **…** *ck*

**Iterate**

*c*l

**Target model**

{

**…**

{

**…**

**…**

**…**

rules to ensure the generated example is seman- tically consistent with the original one and gram- matically correct, such as a synonym dictionary ([Ren et al.](#_bookmark37), [2019](#_bookmark37)), POS checker ([Jin et al.](#_bookmark27), [2019](#_bookmark27)), semantic similarity checker ([Jin et al.](#_bookmark27), [2019](#_bookmark27)), etc. [Alzantot et al.](#_bookmark18) ([2018](#_bookmark18)) applies a traditional language model to score the perturbed sentence at every at- tempt of replacing a word.

These strategies of generating substitutes are un- aware of the context between the substitution po- sitions (usually using language models to test the substitutions), thus are insufﬁcient in ﬂuency con- trol and semantic consistency. More importantly, using language models or POS checkers in scoring the perturbed samples is costly since this trial and error process requires massive inference time.

To overcome the lack of ﬂuency control and se- mantic preservation by using synonyms or simi- lar words in the embedding space, we leverage BERT for word replacement. The genuine na- ture of the masked language model makes sure that the generated sentences are relatively ﬂuent and grammar-correct, also preserve most semantic information, which is later conﬁrmed by human evaluators. Further, compared with previous ap- proaches using rule-based perturbation strategies, the masked language model prediction is context- aware, thus dynamically searches for perturbations rather than simple synonyms replacing.

Different from previous methods using compli- cated strategies to score and constrain the pertur- bations, the contextualized perturbation generator generates minimal perturbations with only one for- ward pass. Without running additional neural mod- els to score the sentence, the time-consuming part is accessing the target model only. Therefore the process is extremely efﬁcient.

**Algorithm 1** BERT-Attack

1: **procedure** WORD IMPORTANCE RANKING

2: *S* = [*w*0*, w*l*,* ] // input: tokenized sentence

*· · ·*

3: *Y* gold-label

*-*

4: **for** *wi* in *S* **do**

5: calculate importance score *Iwi* using Eq. [1](#_bookmark1)

6: select word list *L* = [*wtop-*l*, wtop-*2*, · · ·* ]

7: // sort *S* using *Iwi* in descending order and collect *top . K* words

8: **procedure** REPLACEMENT USING BERT

9: *H* = [*h*0*, , hn*] // sub-word tokenized sequence of *S*

*· · ·*

10: generate top-K candidates for all sub-words using BERT and get *Pen×K*

11: **for** *wj* in *L* **do**

12: **if** *wj* is a whole word **then**

13: get candidate *C* = *F ilter*(*P j*)

14: replace word *wj*

15: **else**

16: get candidate *C* using PPL ranking and Filter

17: replace sub-words [*hj, · · · , hj*＋*t*] 18: Find Possible Adversarial Sample 19: **for** *ck* in C **do**

20: *S/* = [*w*0*, , wj* l*, ck,* ] // attempt

*· · · · · ·-*

21: **if** argmax(*oy*(*S/* ))! = *Y* **then**

22: ***return*** *Sadv* = *S/* // success attack

23: **else**

24: **if** *oy*(*S/* ) *< oy*(*Sadv*) **then**

25: *Sadv* = [*w*0*, · · · , wj-*l*, c, · · ·* ] // do one perturbation

26: ***return* None**

Thus, using the masked language model as a contextualized perturbation generator can be one possible solution to craft high-quality adversarial samples efﬁciently.

## Word Replacement Strategy

As seen in Figure [1](#_bookmark2), given a chosen word *w* to be replaced, we apply BERT to predict the pos- sible words that are similar to *w* yet can mislead the target model. Instead of following the masked language model settings, we do not mask the cho- sen word *w* and use the original sequence as input, which can generate more semantic-consistent sub- stitutes ([Zhou et al.](#_bookmark43), [2019](#_bookmark43)). For instance, given a sequence *”I like the cat.”*, if we mask the word *cat*, it would be very hard for a masked language model to predict the original word *cat* since it could be just as ﬂuent if the sequence is *”I like the dog.”*. Further, if we mask out the given word *w*, for each iteration we would have to rerun the masked lan- guage model prediction process which is costly.

Since BERT uses Bytes-Pair-Encoding (BPE)

to tokenize the sequence *S* = [*w*0*, , wi,* ] into sub-word tokens: *H* = [*h*0*, h*l*, h*2*,* ], we need to align the chosen word to its corresponding sub-words in BERT.

Let denote the BERT model, we feed the tokenized sequence *H* into the BERT to get output prediction *P* = (*H*). Instead of using the argmax prediction, we take the most possible *K* predictions at each position, where *K* is a hyper- parameter.

*M*

*M*

*M*

*· · ·*

*· · · · · ·*

We iterate words that are sorted by word impor- tance ranking process to ﬁnd perturbations. The BERT model uses BPE encoding to construct vo- cabularies. While most words are still single words, rare words are tokenized into sub-words. Therefore, we treat single words and sub-words separately to generate the substitutes.

**Single words** For a single word *wj*, we make attempts using the corresponding top-K predic- tion candidates *Pj*. We ﬁrst ﬁlter out stop words collected from NLTK; for sentiment classiﬁca-

tion tasks we ﬁlter out antonyms using synonym dictionaries ([Mrksˇic´ et al.](#_bookmark33), [2016](#_bookmark33)) since BERT masked language model does not distinguish syn- onyms and antonyms. Then for given candi- date *ck* we construct a perturbed sequence *H/* = [*h*0*, , hj-*l*, ck, hj*＋l ]. If the target model is already fooled to predict incorrectly, we break the loop to obtain the ﬁnal adversarial sample *Hadv*; otherwise, we select from the ﬁltered candidates to pick one best perturbation and turn to the next word in word list *L*.

*· · · · · ·*

**Sub-words** For a word that is tokenized into sub- words in BERT, we cannot obtain its substitutes directly. Thus we use the perplexity of sub-word combinations to ﬁnd suitable word substitutes from predictions in the sub-word level. Given sub-words [*h , h , · · · , h* ] of word *w*, we list all possible

**Yelp** Review classiﬁcation dataset, containing. Following [Zhang et al.](#_bookmark42) ([2015](#_bookmark42)), we process the dataset to construct a polarity classiﬁcation task.

**IMDB** Document-level movie review dataset, where the average sequence length is longer than the Yelp dataset. We process the dataset into a polarity classiﬁcation task [1](#_bookmark4).

*●*

*●*

**AG’s News** Sentence level news-type classi- ﬁcation dataset, containing 4 types of news: World, Sports, Business, and Science.

*●*

**FAKE** Fake News Classiﬁcation dataset, de- tecting whether a news document is fake from Kaggle Fake News Challenge [2](#_bookmark5).

*●*

## Natural Language Inference

* + - * **SNLI** Stanford language inference task ([Bow-](#_bookmark19)

0 l *t*

combinations from the prediction *P*

*et×K*

from *M*,

[man et al.](#_bookmark19), [2015](#_bookmark19)). Given one premise and one

hypothesis, and the goal is to predict if the hy-

which is *Kt* sub-word combinations, we can con- vert them back to normal words by reversing the BERT tokenization process. We feed these combi- nations into the BERT-MLM to get the perplexity of these combinations. Then we rank the perplexity of all combinations to get the top-K combinations to ﬁnd the suitable sub-word combinations.

Given the suitable perturbations, we replace the original word with the most likely perturbation and repeat this process by iterating the importance word ranking list to ﬁnd the ﬁnal adversarial sample. In this way, we acquire the adversarial samples *Sadv* effectively since we only iterate the masked language model once and do perturbations using the masked language model without other checking strategies.

We summarize the two-step BERT-Attack pro- cess in Algorithm [1](#_bookmark3).

# Experiments

## Datasets

We apply our method to attack different types of NLP tasks in the form of text classiﬁcation and natural language inference. Following [Jin et al.](#_bookmark27) ([2019](#_bookmark27)), we evaluate our method on 1k test samples randomly selected from the test set of the given task which are the same splits used by [Alzantot et al.](#_bookmark18) ([2018](#_bookmark18)); [Jin et al.](#_bookmark27) ([2019](#_bookmark27)). The GA method only uses a subset of 50 samples in the FAKE, IMDB dataset.

**Text Classiﬁcation** We use different types of text classiﬁcation tasks to study the effectiveness of our method.

pothesis is entailment, neural, or contradiction of the premise.

**MNLI** Language inference dataset on multi- genre texts, covering transcribed speech, pop- ular ﬁction, and government reports ([Williams](#_bookmark39) [et al.](#_bookmark39), [2018](#_bookmark39)), which is more complicated with diversiﬁed written and spoken style texts, com- pared with the SNLI dataset, including eval data matched with training domains and eval data mismatched with training domains.

*●*

## Automatic Evaluation Metrics

To measure the quality of the generated samples, we set up various automatic evaluation metrics. The success rate, which is the counter-part of after- attack accuracy, is the core metric measuring the success of the attacking method. Meanwhile, the perturbed percentage is also crucial since, gen- erally, less perturbation results in more semantic consistency. Further, under the black-box setting, queries of the target model are the only accessible information. Constant queries for one sample is less applicable. Thus query number per sample is also a key metric. As used in TextFooler ([Jin](#_bookmark27) [et al.](#_bookmark27), [2019](#_bookmark27)), we also use Universal Sentence En- coder ([Cer et al.](#_bookmark20), [2018](#_bookmark20)) to measure the semantic consistency between the adversarial sample and the original sequence. To balance between semantic preservation and attack success rate, we set up a threshold of semantic similarity score to ﬁlter the less similar examples.

1https://datasets.imdbws.com/

2[https://www.kaggle.com/c/f](http://www.kaggle.com/c/fake-news/data)ake-ne[ws/data](http://www.kaggle.com/c/fake-news/data)

**Dataset Method Original Acc Attacked Acc Perturb % Query Number Avg Len Semantic Sim**

BERT-Attack(ours)

**15.5 1.1 1558**

**0.81**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fake** | TextFooler([Jin et al.](#_bookmark27), [2019](#_bookmark27)) | 97.8 19.3 | 11.7 | 4403 885 0.76 | |
|  | GA([Alzantot et al.](#_bookmark18), [2018](#_bookmark18)) | 58.3 | 1.1 | 28508 | - |

BERT-Attack(ours)

**5.1 4.1 273**

**0.77**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Yelp** | TextFooler | 95.6 6.6 | 12.8 | 743 157 0.74 | |
|  | GA | 31.0 | 10.1 | 6137 | - |

BERT-Attack(ours)

**11.4 4.4 454**

**0.86**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IMDB** | TextFooler | 90.9 13.6 | 6.1 | 1134 215 **0.86** | |
|  | GA | 45.7 | 4.9 | 6493 | - |

BERT-Attack(ours)

**10.6 15.4 213**

**0.63**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AG** | TextFooler | 94.2 12.5 | 22.0 | 357 43 | 0.57 |
|  | GA | 51 | 16.9 | 3495 | - |

BERT-Attack(ours)

7.4/**16.1 12.4/9.3 16/30**

0.40/**0.55**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNLI** | TextFooler | 89.4(H/P) **4.0**/20.8 | 18.5/33.4 | 60/142 8/ | 18 **0.45**/0.54 |
|  | GA | 14.7/- | 20.8/- | 613/- | - |

BERT-Attack(ours)

**7.9/11.9 8.8/7.9 19/44**

0.55/**0.68**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| m**M**at**N**ch**L**e**I**d | TextFooler | 85.1(H/P) 9.6/25.3 | 15.2/26.5 | 78/152 11/21 **0.57**/0.65 | |
|  | GA | 21.8/- | 18.2/- | 692/- | - |

BERT-Attack(ours)

**7/13.7 8.0/7.1 24/43**

0.53/**0.69**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mis**M**m**N**at**L**ch**I** ed | TextFooler | 82.1(H/P) 8.3/22.9 | 14.6/24.7 | 86/162 12/22 **0.58**/0.65 | |
|  | GA | 20.9/- | 19.0/- | 737/- | - |

Table 1: Results of attacking against various ﬁne-tuned BERT models. TextFooler is the state-of-the-art baseline. For MNLI task, we attack the hypothesis(H) or premises(P) separately.

## Attacking Results

As shown in Table [1](#_bookmark6), the BERT-Attack method suc- cessfully fool its downstream ﬁne-tuned model. In both text classiﬁcation and natural language infer- ence tasks, the ﬁne-tuned BERTs fail to classify the generated adversarial samples correctly.

The average after-attack accuracy is lower than 10%, indicating that most samples are successfully perturbed to fool the state-of-the-art classiﬁcation models. Meanwhile, the perturb percentage is less than 10 %, which is signiﬁcantly less than previous works.

Further, **BERT-Attack** successfully attacked all tasks listed, which are in diversiﬁed domains such as News classiﬁcation, review classiﬁcation, lan- guage inference in different domains. The results indicate that the attacking method is robust in dif- ferent tasks. Compared with the strong baseline introduced by [Jin et al.](#_bookmark27) ([2019](#_bookmark27))[3](#_bookmark7) and [Alzantot et al.](#_bookmark18) ([2018](#_bookmark18))[4](#_bookmark8), the BERT-Attack method is more efﬁcient

3https://github.com/jind11/TextFooler 4https://github.com/QData/TextAttack

and more imperceptible. The query number and the perturbation percentage of our method are much less.

We can observe that it is generally easier to at- tack the review classiﬁcation task since the perturb percentage is incredibly low. BERT-Attack can mislead the target model by replacing a handful of words only. Since the average sequence length is relatively long, the target model tends to make judg- ments by only a few words in a sequence, which is not the natural way of human prediction. Thus, the perturbation of these keywords would result in in- correct prediction from the target model, revealing the vulnerability of it.

## Human Evaluations

For further evaluation of the generated adversarial samples, we set up human evaluations to measure the quality of the generated samples in ﬂuency and grammar as well as semantic preservation.

We ask human judges to score the grammar cor- rectness of the mixed sentences of generated ad-

versarial samples and original sequences, scoring from 1-5 following [Jin et al.](#_bookmark27) ([2019](#_bookmark27)). Then we ask human judges to make predictions in a shufﬂed mix of original and adversarial texts. We use the IMDB dataset and the MNLI dataset, and for each task, we select 100 samples of both original and adversarial samples for human judges. We ask three human annotators to evaluate the examples. For label pre- diction, we take the majority class as the predicted label, and for semantic and grammar check we use an average score among the annotators.

Seen in Table [2](#_bookmark9), the semantic score and the gram- mar score of the adversarial samples are close to the original ones. MNLI task is a sentence pair prediction task constructed by human crafted hy- potheses based on the premises, therefore original pairs share a considerable amount of same words. Perturbations on these words would make it difﬁ- cult for human judges to predict correctly therefore the accuracy is lower than simple sentence classiﬁ- cation tasks.

**Dataset Accuracy Semantic Grammar**

Original 0.90 3.9 4.0

**MNLI**

Adversarial 0.70 3.7 3.6

Original 0.91 4.1 3.9

**IMDB**

Adversarial 0.85 3.9 3.7

Table 2: Human-Evaluation Results.

## BERT-Attack against Other Models

The BERT-Attack method is also applicable in attacking other target models, not limited to its ﬁne-tuned model only. As seen in Table [3](#_bookmark10), the attack is successful against LSTM-based models, indicating that BERT-Attack is feasible for a wide range of models. Under BERT-Attack, the ESIM model is more robust in the MNLI dataset. We as- sume that encoding two sentences separately gets higher robustness. In attacking BERT-large models, the performance is also excellent, indicating that BERT-Attack is successful in attacking different pre-trained models not only against its own ﬁne- tuned downstream models.

# Ablations and Discussions

## Importance of Candidate Numbers

The candidate pool range is the major hyper- parameter used in the BERT-Attack algorithm. As seen in Figure [2](#_bookmark11), the attack rate is rising along with the candidate size increasing. Intuitively, a larger

**Dataset Model Ori Acc Atk Acc Perturb %**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IMDB** Word-LSTM 89.8 | | | 10.2 | 2.7 |
|  | BERT-Large | 98.2 | 12.4 | 2.9 |
| **Yelp** | Word-LSTM | 96.0 | 1.1 | 4.7 |
|  | BERT-Large | 97.9 | 8.2 | 4.1 |
| **MNLI** | ESIM | 76.2 | 9.6 | 21.7 |
| matched BERT-Large | | 86.4 | 13.2 | 7.4 |

Table 3: BERT-Attack against other models.

100

IMDB

Yelp SNLI

FAKE

MNLI AG

90

Attack success rate

80

70

60

6 12 24 36

K value

Figure 2: Using different candidate number *K* in the attacking process.

*K* would result in less semantic similarity. How- ever, the semantic measure via Universal Sentence Encoder is maintained in a stable range, (experi- ments show that semantic similarities drop less than 2%), indicating that the candidates are all reason- able and semantically consistent with the original sentence.

Further, a ﬁxed candidate number could be rigid in practical usage, so we run a test using a threshold to cut off candidates that are less possible as a plausible perturbation.

As seen in Table [4](#_bookmark12), when using a ﬂexible thresh- old to cut off unsuitable candidates, the attacking process has a lower query number. This indicates that some candidates predicted by the masked lan- guage model with a lower prediction score may not be meaningful so skipping these candidates can save the unnecessary queries.

**Dataset Method Ori Acc Atk Acc Queries % IMDB** Fixed-*K* 90.9 11.4 454

With Threshold 90.9 12.4 440

Table 4: Flexible Candidates Using a threshold to cut off unsuitable candidates.

## 5.2 Importance of Sequence Length

The BERT-Attack method is based on the contextu- alized masked language model. Thus the sequence length plays an important role in the high-quality perturbation process. As seen, instead of the previ- ous methods focusing on attacking the hypothesis of the NLI task, we aim at premises whose aver- age length is longer. This is because we believe that contextual replacement would be less reason- able when dealing with extremely short sequences. To avoid such a problem, we believe that many word-level synonym replacement strategies can be combined with BERT-Attack, allowing the BERT- Attack method to be more applicable.

|  |  |  |
| --- | --- | --- |
| BERT-Atk 85.1 | 7.9 | 8.8 |
| matched +Adv Train 84.6 | 23.1 | 10.5 |

**Dataset Method Ori Acc Atk Acc Perturb % MNLI**

Table 5: Adversarial training results.

**Dataset Model** LSTM BERT-base BERT-large Word-LSTM - 0.78 0.75

indicating that the model is harder to be attacked. Therefore, the generated dataset can be used as additional data for further exploration of making neural models more robust.

**Dataset Model Atk Acc Perturb % Semantic Yelp** BERT-Atk 5.1 4.1 0.77

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | w/o sub-word | 7.1 | 4.3 | 0.74 |
| **MNLI** | BERT-Atk | 11.9 | 7.9 | 0.68 |
|  | w/o sub-word | 14.7 | 9.3 | 0.63 |

Table 7: Effects on sub-word level attack.

## 5.4 Effects on Sub-Word Level Attack

BPE method is currently the most efﬁcient way to deal with a large number of words, as used in BERT. We establish a comparative experiment where we do not use the sub-word level attack. That is we skip those words that are tokenized with multiple sub-words.

As seen in Table [7](#_bookmark15), using the sub-word level attack can achieve higher performances, not only in higher attacking success rate but also in less

**IMDB**

BERT-base 0.83 - 0.71

BERT-large 0.87 0.86 -

perturbation percentage.

**Dataset Method Atk Acc Perturb % Semantic**

**Dataset Model** ESIM BERT-base BERT-large

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ESIM | - | 0.59 | 0.60 |
| **MNLI** | BERT-base | 0.60 | - | 0.45 |
|  | BERT-large | 0.59 | 0.43 | - |

**MNLI**

matched

MIR 7.9 8.8 0.68

Random 20.2 12.2 0.60

LIR 27.2 15.0 0.60

Table 6: Transferability analysis using attacked accu- racy as the evaluation metric. The column is the target model used in attack, and the row is the tested model.

## 5.3 Transferability and Adversarial Training

To test the transferability of the generated adver- sarial samples, we take samples aimed at different target models to attack other target models. Here, we use BERT-base as the masked language model for all different target models. As seen in Table [6](#_bookmark14), samples are transferable in NLI task while less transferable in text classiﬁcation.

Meanwhile, we further ﬁne-tune the target model using the generated adversarial samples from the train set and then test it on the test set used before. As seen in Table [5](#_bookmark13), generated samples used in ﬁne- tuning help the target model become more robust while accuracy is close to the model trained with clean datasets. The attack becomes more difﬁcult,

Table 8: Most Importance Ranking (MIR) vs Least Im- portance Ranking (LIR)

## Effects on Word Importance Ranking

Word importance ranking strategy is supposed to ﬁnd keys that are essential to NN models, which is very much like calculating the maximum risk of wrong predictions in the FGSM algorithm ([Good-](#_bookmark25) [fellow et al.](#_bookmark25), [2014](#_bookmark25)). When not using word im- portance ranking, the attacking algorithm is less successful.

**Dataset Method Runtime(s/sample)**

BERT-Attack(w/o BPE) 14.2

**IMDB** BERT-Attack(w/ BPE) 16.0

Textfooler([Jin et al.](#_bookmark27), [2019](#_bookmark27)) 42.4

GA([Alzantot et al.](#_bookmark18), [2018](#_bookmark18)) 2582.0

Table 9: Runtime comparison.

**Dataset** Label

Ori Some rooms have balconies . Hypothesis All of the rooms have balconies off of them . Contradiction

**MNLI**

Adv Many rooms have balconies . Hypothesis All of the rooms have balconies off of them . Neutral it is hard for a lover of the novel northanger abbey to sit through this bbc adaptation and to Negative

**IMDB**

Ori

keep from throwing objects at the tv screen... why are so many facts concerning the tilney family and mrs . tilney ’ s death altered unnecessarily ? to make the story more ‘ horrible ? ’

it is hard for a lover of the novel northanger abbey to sit through this bbc adaptation and to Positive

Adv keep from throwing objects at the tv screen... why are so many facts concerning the tilney family and mrs . tilney ’ s death altered unnecessarily ? to make the plot more ‘ horrible ? ’

i ﬁrst seen this movie in the early 80s .. it really had nice picture quality too . anyways , i ’m Positive

**IMDB**

Ori

glad i found this movie again ... the part i loved best was when he hijacked the car from this poor guy... this is a movie i could watch over and over again . i highly recommend it .

i ﬁrst seen this movie in the early 80s .. it really had nice picture quality too . anyways , i ’m Negative

Adv glad i found this movie again ... the part i loved best was when he hijacked the car from this poor guy... this is a movie i could watch over and over again . i inordinately recommend it .

Table 10: Some generated adversarial samples. Origin label is the correct prediction while label is adverse predic- tion. Only red color parts are perturbed. We only attack premises in MNLI task. Text in FAKE dataset and IMDB dataset is cut to ﬁt in the table. Original text contains more than 200 words.

## Runtime Comparison

Since BERT-Attack does not use language mod- els or sentence encoders to measure the output se- quence during the generation process, also, the query number is lower, therefore the runtime is faster than previous methods. As seen in Table [9](#_bookmark16), BERT-Attack is much faster than generic algo- rithm ([Alzantot et al.](#_bookmark18), [2018](#_bookmark18)) and 3 times faster then Textfooler.

## Examples of Generated Adversarial Sentences

As seen in Table [10](#_bookmark17), the generated adversarial sam- ples are semantically consistent with its original input, while the target model makes incorrect pre- dictions. In both review classiﬁcation samples and language inference samples, the perturbations do not mislead human judges.

# Conclusion

In this work, we propose a high-quality and effec- tive method **BERT-Attack** to generate adversarial samples using BERT masked language model. Ex- periment results show that the proposed method achieves a high success rate while maintaining a minimum perturbation. Nevertheless, candidates generated from the masked language model can sometimes be antonyms or irrelevant to the original words, causing a semantic loss. Thus, enhancing language models to generate more semantically re- lated perturbations can be one possible solution to perfect BERT-Attack in the future.

# Acknowledgments

We would like to thank the anonymous review- ers for their valuable comments. We are thank- ful for the help of Demin Song, Hang Yan and Pengfei Liu. This work was supported by the Na- tional Natural Science Foundation of China (No. 61751201, 62022027 and 61976056), Shanghai

Municipal Science and Technology Major Project (No. 2018SHZDZX01) and ZJLab.

# References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. [Generating natural language adversar-](http://arxiv.org/abs/1804.07998) [ial examples](http://arxiv.org/abs/1804.07998). *CoRR*, abs/1804.07998.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large anno- tated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anu- pam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of](http://arxiv.org/abs/1810.04805) [deep bidirectional transformers for language under-](http://arxiv.org/abs/1810.04805) [standing](http://arxiv.org/abs/1810.04805). *CoRR*, abs/1810.04805.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yan- jun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classiﬁers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversar- ial examples. *arXiv preprint arXiv:1412.6572*.

Robin Jia and Percy Liang. 2017. Adversarial exam- ples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. [Is BERT really robust? natural](http://arxiv.org/abs/1907.11932) [language attack on text classiﬁcation and entailment](http://arxiv.org/abs/1907.11932). *CoRR*, abs/1907.11932.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Di- makis, Inderjit S Dhillon, and Michael Witbrock. 2019. Discrete adversarial attacks and submodular optimization with applications to text classiﬁcation. *Systems and Machine Learning (SysML)*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classiﬁcation can be fooled. *arXiv preprint arXiv:1704.08006*.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. 2017. Virtual adversarial training: A regularization method for supervised and semi- supervised learning. volume 41, pages 1979–1993.

Nikola Mrksˇic´, Diarmuid O Se´aghdha, Blaise Thom- son, Milica Gasˇic´, Lina Rojas-Barahona, Pei- Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-ﬁtting word vec- tors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High con- ﬁdence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vi- sion and pattern recognition*, pages 427–436.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1532–1543.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lip- ton. 2019. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial ex- amples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial trig- gers for attacking and analyzing NLP. *Empirical Methods in Natural Language Processing*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sen- tence understanding through inference. In *Proceed- ings of the Conference of the North American Chap- ter of the Association for Computational Linguistics: Human Language Technologies*, pages 1112–1122.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. 2018. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combina- torial optimization. In *Proceedings of the 58th An- nual Meeting of the Association for Computational Linguistics*, pages 6066–6080.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text clas- siﬁcation. In *Advances in neural information pro- cessing systems*, pages 649–657.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. [BERT-based lexical substitution](https://doi.org/10.18653/v1/P19-1328). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Compu- tational Linguistics.